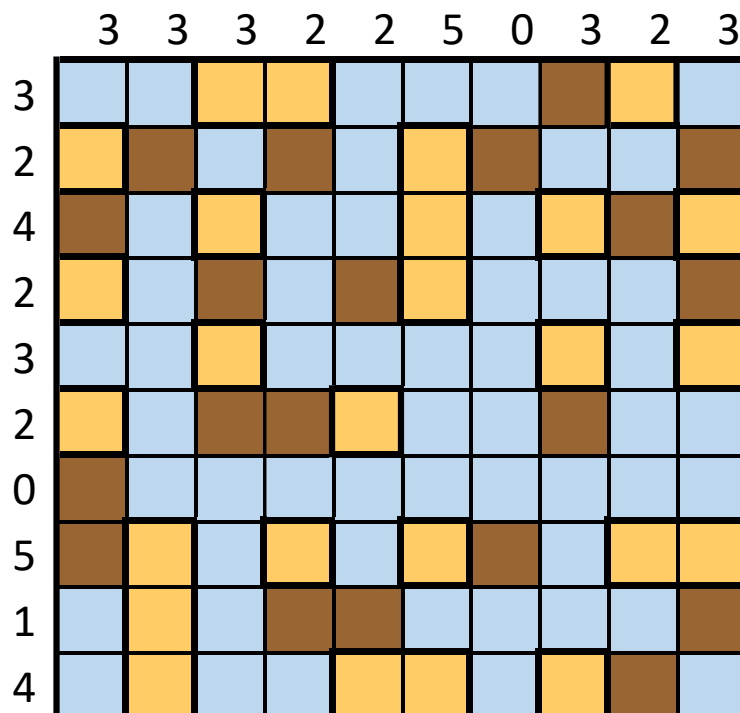


Programação Orientada por Objetos

Boats & Docks



Ano Letivo: 2020/2021

Projeto de Época Normal e de Recurso – 1ª e 2ª Fase

Índice

1.	Introdução.....	1
2.	Funcionalidades	2
2.1.	O Jogo.....	2
2.1.1.	Casas de jogo.....	2
2.1.2.	Regras.....	5
2.1.3.	Jogadas.....	7
2.1.4.	Níveis.....	9
2.1.5.	Sistema de pontuação.....	10
2.2.	Jogadores	11
2.3.	Editor de níveis.....	12
2.4.	Desenrolar do jogo em modo consola	13
	Menus do jogo	17
3.	Considerações.....	18
3.1.	Fases de desenvolvimento e entrega	18
3.2.	Implementação e codificação	19
3.3.	Constituição de grupos	19
4.	Entrega do projeto	19
5.	Regras e Critérios de Avaliação do Projeto.....	21
5.1.	Regras de Avaliação	21
5.2.	Critérios de Avaliação	21
5.2.1.	1ª Fase.....	21
5.2.2.	2ª Fase.....	22
6.	Resumo das Datas Importantes	23
6.1.	Entrega da 1ª fase	23
6.2.	Entrega da 2ª fase	23

1. Introdução

O Projeto final pretende que o aluno sistematize os conhecimentos adquiridos de maneira a estruturar corretamente o código desenvolvido tornando-o eficiente, robusto, reutilizável e modular, para assim poder construir uma solução aplicacional de acordo com o paradigma orientado por objetos, utilizando a linguagem de programação java. O objetivo passa pelo desenvolvimento de uma versão de um jogo de tabuleiro chamado **Boats & Docks**.

O jogo **Boats & Docks** é um jogo tradicional, jogado individualmente, cuja lógica de jogo consiste numa mistura de regras entre o jogo **Minesweeper** e o **Sudoku**, sendo o principal objetivo do jogo atracar todos os barcos nos respetivos portos, seguindo um conjunto de regras. Podem consultar uma versão jogável de um jogo ([Android](#), [IOS](#)) cuja dinâmica é semelhante. Contudo, as regras do jogo a desenvolver neste projeto divergem em diversos pontos.

O projeto será desenvolvido em três fases distintas. A primeira fase consiste na análise e desenho da aplicação, resultante da implementação das classes de domínio. A segunda fase tem foco na implementação das funcionalidades e lógica de jogo, em ambiente de consola. Por último, a terceira fase será dedicada à criação da interface gráfica com o utilizador.

Tenha particular atenção ao uso do paradigma de POO na modelação das classes, i.e., ao correto uso dos conceitos de encapsulamento, herança, classes abstratas, polimorfismo, interfaces, maximização da coesão (responsabilidade única), minimização do acoplamento, desenho orientado por responsabilidades, etc. Uma modelação bem pensada facilitará todo o processo de desenvolvimento e manutenção.

O presente documento versa apenas a primeira e segunda fase do projeto. Em data oportuna, após entrega de ambas as fases, será apresentado o descritivo da terceira fase. O desenvolvimento é iterativo, pressupondo o melhoramento contínuo de componentes implementadas em fases anteriores.

2. Funcionalidades

Descrevem-se, nesta secção, um conjunto de funcionalidades, consideradas como requisitos funcionais, com os quais o projeto implementado deverá cumprir.

2.1. O Jogo

O objetivo deste jogo passa por identificar todas as casas do tabuleiro de jogo, de forma a obter a maior pontuação possível, por nível. Quando o jogo inicia, apresenta todas as casas do tipo **porto**, cabe ao jogador determinar quais as casas do tipo **água** e **barco**. Existem três dimensões de barcos diferentes, sendo estes **pequenos**, **médios** e **grandes**. O jogo termina quando cada porto tiver um barco atracado, seguindo as respetivas regras.

Apenas quando colocados todos os barcos, o jogador pode validar o tabuleiro. Quando validado, todas as casas **desconhecidas** passam automaticamente ao estado **água**. Se o tabuleiro estiver correto, o jogo termina e é calculada a pontuação obtida. Caso contrário, o jogador é informado do número de erros existentes no tabuleiro.

As casas erradas no momento da validação do tabuleiro de jogo, são declaradas como falhadas. Uma casa, uma vez falhada já não voltará a alterar o seu estado. Se a casa falhada for do tipo barco, as restantes casas que compõem o barco (caso existam), devem ser marcadas igualmente como falhadas. No final do jogo, é contabilizado o número total de casas falhadas e esse valor reflete uma penalização na pontuação do nível.

2.1.1. Casas de jogo

Neste jogo, existem duas tipologias de casa de jogo:

- Casas estáticas (inalteráveis)
 - **Porto** – identifica um porto marítimo, adjacente ao qual irá ser atracado um barco. Em cada nível são dadas a conhecer no início do jogo e não podem ser alteradas.

- Casas interativas (alteráveis) – o jogador, ao clicar nesta casa, altera o seu tipo entre a seguinte lista:

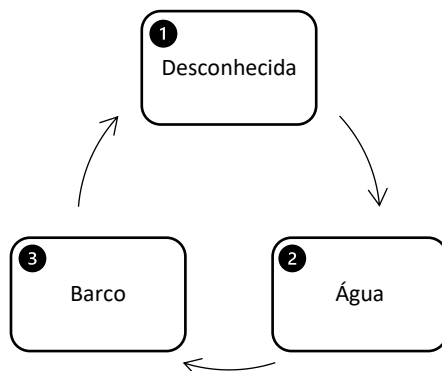


Figura 1 - Sequência da alteração de estado de uma casa interativa

1. **Desconhecida** – no início do jogo, todas as casas do tabuleiro que não são do tipo “porto”, são do tipo “desconhecida”. Esta representa uma casa neutra, cujo tipo ainda não foi identificado pelo jogador. Uma casa identificada como “desconhecida”, não contará para efeitos de validação de um tabuleiro de jogo. Um jogador, ao clicar numa casa deste tipo, irá alterar a sua tipologia para o tipo “água”.
2. **Água** – Esta casa representa um setor com água no tabuleiro de jogo. Esta casa fornece ao jogador dois pontos quando corretamente colocada. Um jogador, ao clicar numa casa deste tipo, altera a sua tipologia para o tipo “barco”.
3. **Barco** – Esta casa representa um barco, tem de ser colocada sempre adjacente a uma casa do tipo “porto”, segundo as regras do jogo. Esta casa fornece ao jogador cinco pontos quando corretamente colocada. Ao clicar nesta casa, volta ao tipo “desconhecida”. Podem existir barcos de três tamanhos distintos:
 - **Barco Pequeno** – composto por apenas uma casa, totaliza cinco pontos quando corretamente colocada.
 - **Barco Médio** - composto por duas casas, totaliza dez pontos quando corretamente colocadas.
 - **Barco Grande** - composto por três casas, totaliza quinze pontos quando corretamente colocadas.

Independentemente do tamanho do barco, este tem de estar atracado a um porto. Isto é, adjacente, quer vertical, quer horizontalmente, conforme ilustrado na figura seguinte:

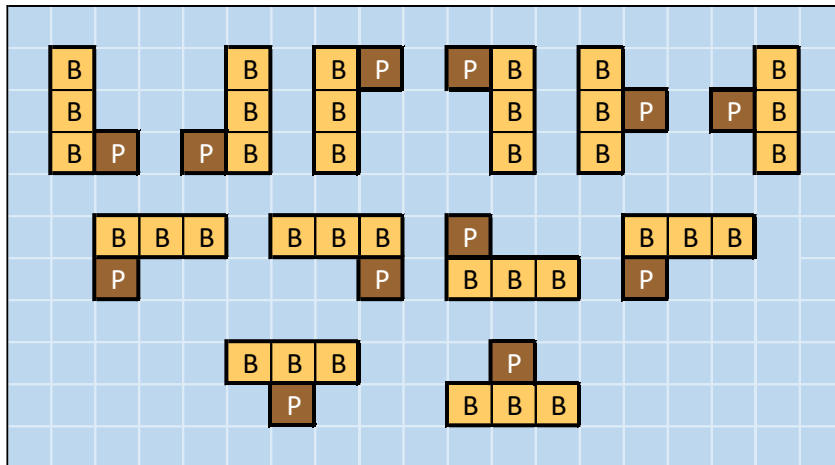


Figura 2 - Exemplos de correta colocação de um barco grande, relativamente ao porto

No entanto, este nunca pode ser colocado perpendicular ao respetivo porto. Esta regra não se aplica aos barcos pequenos, com apenas uma casa.

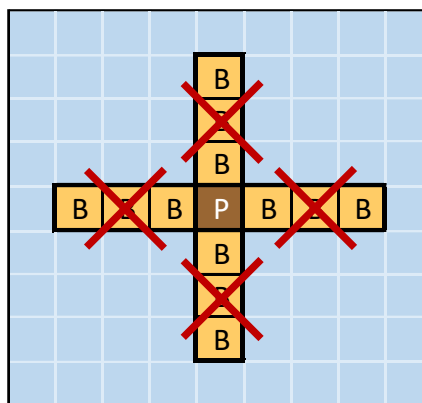


Figura 3 - Exemplo de incorreta colocação de um barco grande, relativamente a um porto

Após dispor todas as casas no tabuleiro de jogo, é possível validar a correta colocação das mesmas. Uma casa indevidamente identificada, deve ser marcada como falhada. Ainda que possa ser alterada, de forma que o jogador possa completar o jogo, a marcação da falha deve permanecer inalterável para efeitos de pontuação. Relembrar que, quando a casa falhada é do tipo barco, as restantes casas que o compõem serão igualmente marcadas como falhadas. Assim, quando falhada uma casa do tipo barco, este deixa de contar para bonificação na pontuação final.

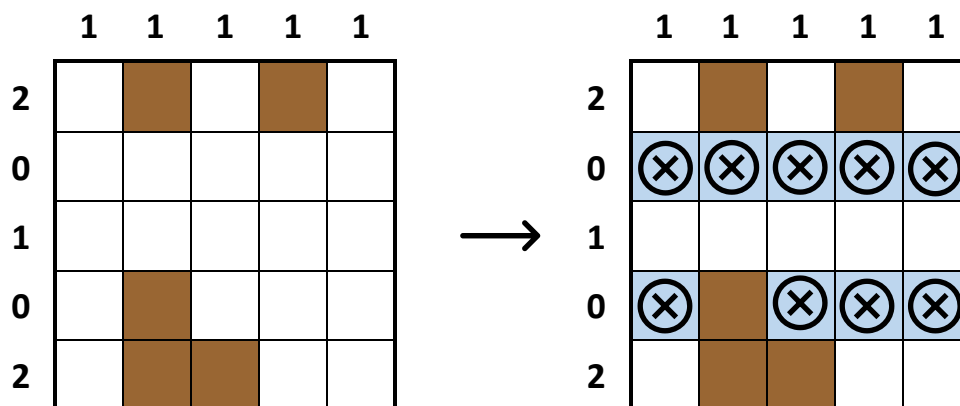
2.1.2. Regras

Para atracar os barcos nos respetivos portos, é necessário obedecer a um conjunto de regras. No início do jogo, todos os portos existentes são dispersos num tabuleiro. Então, cada barco irá atracar num porto, e todos os portos terão um e só um barco. As restantes casas, serão água.

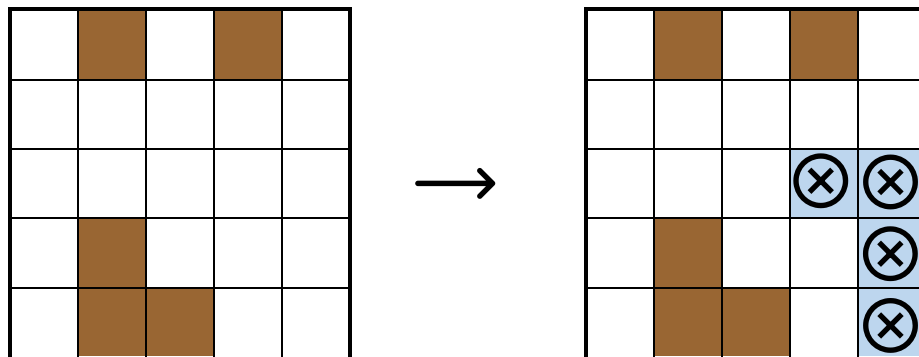
Os números à volta do tabuleiro identificam o número de barcos a colocar, na linha (horizontal) ou coluna (vertical) correspondente. Cada barco deve estar numa das quatro casas adjacentes ao porto em questão (verticalmente ou horizontalmente, mas não diagonalmente). Os barcos não se podem tocar, nem mesmo diagonalmente.

Estratégia e Dicas

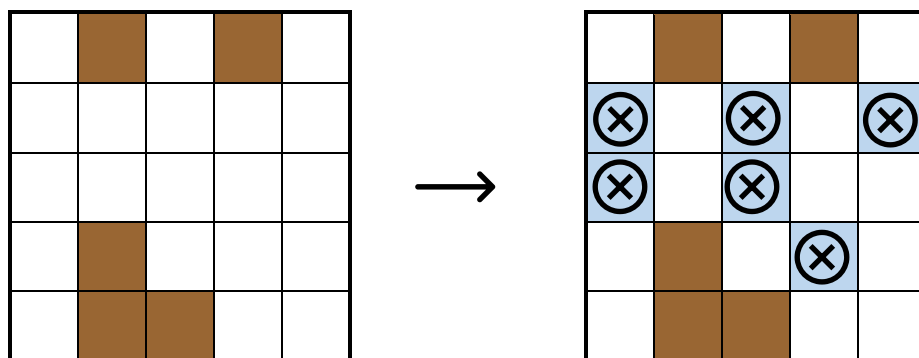
- **Linhas e/ou colunas com número 0:** Estude o tabuleiro cuidadosamente e procure as linhas e/ou colunas com o número 0. Elas **não** contêm nenhum barco, pelo que poderá marcar todas as casas dessas linhas e/ou colunas, como sendo água.



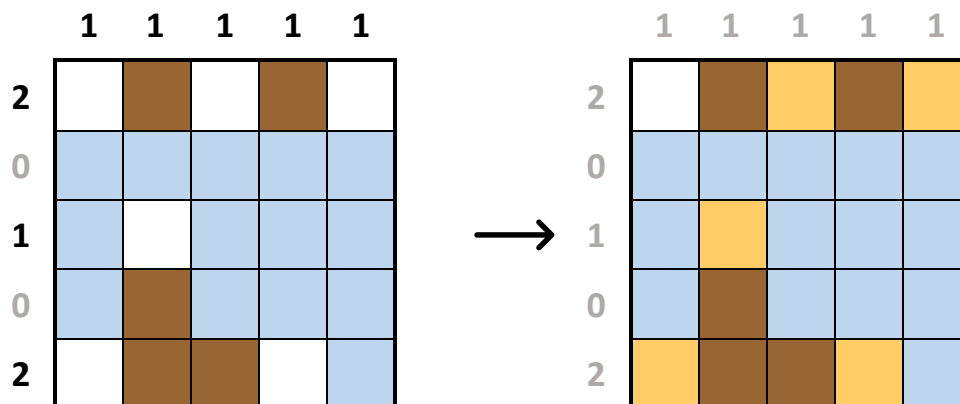
- **Casas muito longe de um porto:** As casas que estão muito distantes de um porto (a mais de uma casa de espaçamento) não devem conter um barco, pelo que poderá identificá-las como água.



- **Casas diagonais a um porto:** Algumas casas encontram-se na diagonal relativamente a um porto. Se essa casa, para além de ser diagonal de um porto, estiver também horizontal ou verticalmente adjacente a outro porto, pode conter um barco. Caso contrário, pode descartar a hipótese de atracar um barco nestas casas.

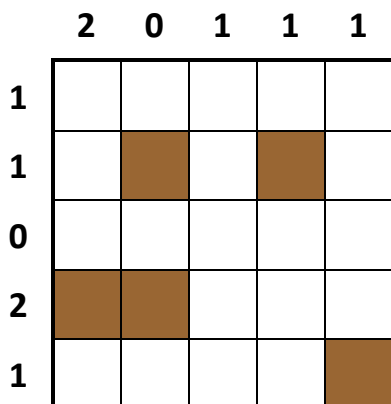


- **Número de casas igual aos números no topo do tabuleiro:** Em linhas e/ou colunas onde o número corresponde ao número de casas livres, pode colocar um barco em todas as casas livres.



2.1.3. Jogadas

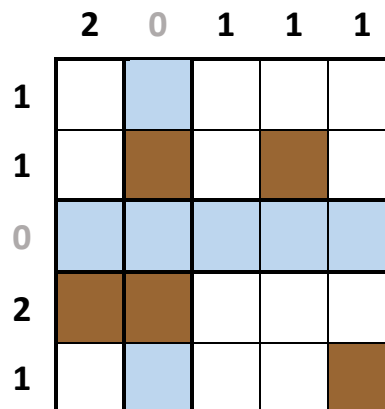
Segue-se um exemplo detalhado, do pensamento de resolução de um determinado nível do jogo **Boats & Docks**.



Jogada 1: O nível em análise tem um grau de dificuldade “fácil”, pelo que se estima um tempo de conclusão de 30 segundos.

No início do jogo, é apresentado ao jogador o respetivo tabuleiro, com a representação de todos os portos existentes. Nas colunas/linhas adjacentes ao tabuleiro, o jogador consegue observar quantos barcos existirão nas colunas/linhas.

Jogada 2: O jogador começa por identificar todas as linhas e colunas em que não poderão ser colocadas nenhuma casa do tipo barco. Desta forma, essas mesmas casas podem ser marcadas como sendo do tipo água, descartando um conjunto de possibilidades.



	2	0	1	1	1
1	blue	blue	white	white	white
1	yellow	brown	white	brown	white
0	blue	blue	blue	blue	blue
2	brown	brown	white	white	white
1	yellow	blue	white	white	brown

Jogada 3: Sabendo que os barcos têm de ser dispostos (vertical ou horizontalmente) adjacentes a um porto, o jogador, ao observar a primeira coluna, verifica logo a existência de duas casas de jogo que serão ocupadas por um barco. A casa sobranete dessa mesma coluna, por sua vez, e tendo sido o número total de barcos da coluna colocados, será identificada como água.

Jogada 4: Tendo colocado estes dois barcos, o jogador verifica que, nessas linhas, totalizou o número de barcos possíveis. Dessa forma, sabe automaticamente que as restantes casas das linhas serão do tipo água.

	2	0	1	1	1
1	blue	blue	white	white	white
1	yellow	brown	blue	brown	blue
0	blue	blue	blue	blue	blue
2	brown	brown	white	white	white
1	yellow	blue	blue	blue	brown

	2	0	1	1	1
1	blue	blue	white	yellow	white
1	yellow	brown	blue	brown	blue
0	blue	blue	blue	blue	blue
2	brown	brown	white	blue	white
1	yellow	blue	blue	blue	brown

Jogada 5: De seguida, verifica a existência de um porto onde, sabendo que nenhum barco pode estar numa casa diagonal ao respetivo porto, apenas tem uma possibilidade de colocação do barco. Sendo completada a respetiva coluna, a casa sobranete é identificada como água.

Jogada 6: Restando a colocação de dois barcos para apenas dois portos livres, observando o tabuleiro, o jogador identifica existirem apenas duas casas nessas condições, pelo que os barcos podem ser colocados.

	2	0	1	1	1
1					
1					
0					
2					
1					

	2	0	1	1	1
1					
1					
0					
2					
1					

Jogada 7: Por último, tendo sido colocados todos os barcos, as casas sobrantes serão preenchidas como água, totalizando assim a colocação de peças no tabuleiro. O jogo terminou.

2.1.4. Níveis

Cada tabuleiro de jogo corresponde a um nível diferente. Cada nível, pode ter um dos seguintes graus de dificuldade:

Tabela 1 - Graus de dificuldade e respetivos parâmetros

Grau de Dificuldade	Dimensão do tabuleiro	Tempo expectável de resolução	Pontuação base
Fácil	5x5	30 segundos	50 pontos
Intermédio	7x7	90 segundos	100 pontos
Difícil	10x10	150 segundos	150 pontos

Para além do tabuleiro de jogo, o nível deverá também, conter informação relativa ao identificador do nível (nome pelo qual deverá ser identificado), autor (quem criou) e a respetiva data de criação.

Mediante o grau de dificuldade do nível, é determinado o tipo de barcos existentes e a respetiva quantidade, conforme observado na seguinte tabela.

Tabela 2 - Tipos de barco por grau de dificuldade

Grau de Dificuldade	Barco Pequeno	Barco Médio	Barco Grande
Fácil	Até 5	-	-
Intermédio	Até 7	3	-
Difícil	Até 12	3	2

2.1.5. Sistema de pontuação

Apresenta-se, de seguida, o sistema de pontuação, sendo que todas as pontuações obtidas devem ser persistidas (2ª fase) de forma a manter todo o histórico de jogo.

- **Pontuação por Nível**

Atinge a maior pontuação, o jogador que completar o nível mais rápido, com menor número de tentativas e menor número de casas falhadas. A cada nível é associada uma pontuação base, definido pelo grau de dificuldade do mesmo (e.g, um nível de dificuldade “fácil”, tem um tempo estimado de conclusão de 30 segundos e uma pontuação base de 50 pontos).

Quando completado o tabuleiro, o mesmo pode ser validado. Cada validação corresponde a uma tentativa, sendo que mais do que uma tentativa por nível implica uma penalização de 10 pontos. No entanto, um jogo completo sem qualquer falha, isto é, validado à primeira tentativa, incorre de uma bonificação de 20 pontos.

A contagem do tempo inicia no momento da colocação da primeira casa de jogo, terminando apenas quando todo o tabuleiro estiver corretamente preenchido. O tempo total de jogo será comparado ao tempo de base definido para o nível em questão. Caso o tempo de jogo ultrapasse o tempo base, constitui uma penalização. Por outro lado, quando o tempo de jogo for inferior ao tempo base, constitui uma bonificação.

Tabela 3 – Pontuação

Penalizações	
Falha na casa validada (água)	-2 pontos por cada casa falhada
Falha na casa validada (barco)	-5 pontos por cada segmento do barco falhado
Mais do que uma tentativa de validação	-10 pontos por cada validação a mais
Tempo de jogo para além do tempo base	-1 pontos por cada segundo
Bonificações	
Apenas uma validação	+20 pontos
Tempo de jogo inferior ao tempo base	+1 pontos por cada segundo

De uma forma geral, a pontuação final traduz-se na seguinte (simples) equação, sendo que no extremo, o valor resultante não poderá ser negativo:

$$\text{Pontuação final} = \text{pontuação base} - \text{penalizações} + \text{bonificações}$$

No final do jogo, o jogador é informado se foi atingido uma nova pontuação máxima, relativamente ao nível que completou. Caso contrário, juntamente com a pontuação obtida, é apresentada a sua melhor pontuação já alcançada, até ao momento. No caso de obter uma pontuação máxima, esta será registada como a pontuação atual, obtida pelo jogador, para o nível em questão.

- **Pontuação Acumulada**

Cada jogador tem uma pontuação acumulada, resultante da soma das melhores pontuações atingidas, em cada nível jogado. Esta pontuação será alterada cada vez que se atingir uma pontuação máxima num determinado nível.

2.2. Jogadores

O jogo é jogado individualmente, por um jogador. Este jogador é identificado por um nome (*nickname*), introduzido pelo próprio jogador, que deverá ser único. Cada jogador deve manter um histórico de pontuações obtidas, para todos os jogos efetuados pelo próprio, com registo de data e hora do jogo. Deve ser possível aceder à pontuação máxima obtida pelo jogador, em cada um dos níveis. Deve ainda, ser apresentado um ranking, onde apresenta todos os jogadores, ordenados de forma decrescente, com base na pontuação acumulada de cada.

2.3. Editor de níveis

O utilizador poderá criar novos níveis. Para tal, deve determinar o grau de dificuldade do nível a ser criado. Mediante o grau de dificuldade, é determinado o **número total de portos possíveis** de colocar no tabuleiro: Fácil (5), Médio (10) e Difícil (17).

As regras de validação do tabuleiro criado seguem as mesmas regras da validação do tabuleiro em momento de jogo, isto é, apos a criação de um novo nível é necessário verificar se este não contém erros.

Para a criação do tabuleiro, o utilizador deve apenas especificar quais as casas do tipo **porto** e **barco**. Todas as restantes serão do tipo **água**.

Para marcação das respetivas casas no tabuleiro, sugere-se a leitura de *inputs* no formato:

<Posição horizontal>,<posição vertical>—<tipo de casa>

Como por exemplo:

0,2-B (para a criação da casa do tipo barco na posição 0,2 do tabuleiro)

1,2-P (para a criação da casa do tipo porto na posição 1,2 do tabuleiro)

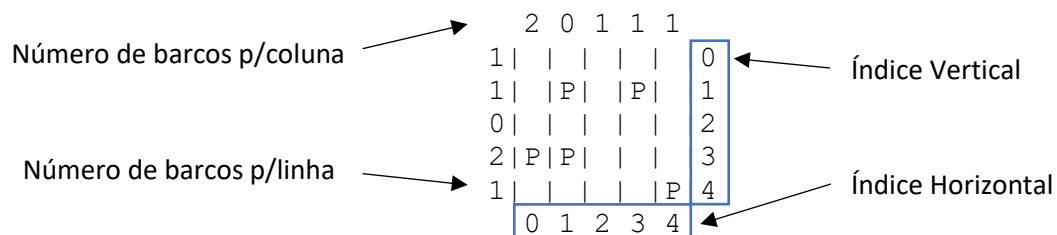
Quando terminada a inserção das casas de jogo no respetivo tabuleiro, deve ser possível armazenar a informação num ficheiro, cujo conteúdo será idêntico ao apresentado em seguida:

```
..BB...PB.
...P.BP...P
P.B...B.BPB
B.P.PB...P
.....B.B
B..PB..P..
P.....
PB.B.BP.BB
.B.PP....P
.B..BB.BP.
#Nível 1
#POO
#10/03/2021
```

Este ficheiro irá conter a informação do nível criado, nomeadamente o tabuleiro de jogo, o identificador do nível (nome), autor e a data de criação. O utilizador pode definir a localização onde pretende persistir esta informação. O ficheiro deverá ter a extensão **.lvl** e seguir a nomenclatura **BoatsAndDocks_<identificador>.<extensão>**. Desta forma, o respetivo nível poderá ser jogado conjuntamente com os restantes níveis fornecidos.

2.4. Desenrolar do jogo em modo consola

De seguida, representa-se uma proposta de solução para a implementação do jogo em modo consola. O jogador deverá ir seleccionando a casa na qual pretende modificar o estado, através da coordenada (horizontal, vertical) em que a casa se situa, no tabuleiro de jogo. Conforme mencionado nas regras de jogo, o tabuleiro inicia-se com as casas estáticas do tipo porto ("P") inseridas no tabuleiro de jogo, as restantes, inicialmente desconhecidas, são alteráveis pela ordem: desconhecida (" "), água (".") e barco ("B").



Inicia-se, de seguida, uma simulação de jogo em modo consola:

```

      2 0 1 1 1
      1| | | | |0
      1| |P| |P| |1
      0| | | | |2
      2|P|P| | |3
      1| | | | |P4
      0 1 2 3 4

Indique a próxima jogada (Horizontal,Vertical): 0,2

```

Para jogar o jogador deve indicar a posição da casa que pretende alterar. Neste caso, o jogador começou por marcar casas do tipo água nas posições onde sabe não existirem barcos, vejamos:

```

      2 0 1 1 1
      1| | | | |0
      1| |P| |P| |1
      0|. | | | |2
      2|P|P| | |3
      1| | | | |P4
      0 1 2 3 4

Indique a próxima jogada (Horizontal,Vertical): 1,2

```



```

      2 0 1 1 1
1| | | | |0
1| |P| |P| |1
0|.|.|.|.|.2
2|P|P| | |3
1| | | | |P4
      0 1 2 3 4

```

Indique a próxima jogada (Horizontal,Vertical): **2,2**

```

      2 0 1 1 1
1| | | | |0
1| |P| |P| |1
0|.|.|.|.|.2
2|P|P| | |3
1| | | | |P4
      0 1 2 3 4

```

Indique a próxima jogada (Horizontal,Vertical): **3,2**

```

      2 0 1 1 1
1| | | | |0
1| |P| |P| |1
0|.|.|.|.|.2
2|P|P| | |3
1| | | | |P4
      0 1 2 3 4

```

Indique a próxima jogada (Horizontal,Vertical): **4,2**

```

      2 0 1 1 1
1| | | | |0
1| |P| |P| |1
0|.|.|.|.|.2
2|P|P| | |3
1| | | | |P4
      0 1 2 3 4

```

Indique a próxima jogada (Horizontal,Vertical): **1,0**

Neste momento, o jogador completou a primeira linha totalmente preenchida, onde não havia a possibilidade de colocação de barcos, por isso, todas as casas são do tipo água. Existe uma coluna nesta mesma condição, pelo que será preenchida em seguida.

```

      2 0 1 1 1
1| |.| | | |0
1| |P| |P| |1
0|.|.|.|.|.|2
2|P|P| | | |3
1| | | | |P|4
    0 1 2 3 4

```

Indique a próxima jogada (Horizontal,Vertical): **1,4**

```

      2 0 1 1 1
1| |.| | | |0
1| |P| |P| |1
0|.|.|.|.|.|2
2|P|P| | | |3
1| |.| | |P|4
    0 1 2 3 4

```

Indique a próxima jogada (Horizontal,Vertical): **0,4**

Preenchidas todas as linhas e colunas que não contenham barcos, estamos em condições de colocar os primeiros barcos no tabuleiro. Uma casa, para passar ao estado barco, deve ser selecionada duas vezes.

```

      2 0 1 1 1
1| |.| | | |0
1| |P| |P| |1
0|.|.|.|.|.|2
2|P|P| | | |3
1|.|.| | |P|4
    0 1 2 3 4

```

Indique a próxima jogada (Horizontal,Vertical): **0,4**

```

      2 0 1 1 1
1| |.| | | |0
1| |P| |P| |1
0|.|.|.|.|.|2
2|P|P| | | |3
1|B|.| | |P|4
    0 1 2 3 4

```

Indique a próxima jogada (Horizontal,Vertical): **0,1**

```

      2 0 1 1 1
1| |.| | | |0
1|.|P| |P| |1
0|.|.|.|.|2
2|P|P| | |3
1|B|.| | |P4
      0 1 2 3 4

```

Indique a próxima jogada (Horizontal,Vertical): **0,1**

```

      2 0 1 1 1
1| |.| | | |0
1|B|P| |P| |1
0|.|.|.|.|2
2|P|P| | |3
1|B|.| | |P4
      0 1 2 3 4

```

Indique a próxima jogada (Horizontal,Vertical): **0,0**

```

      2 0 1 1 1
1|.|.| | | |0
1|B|P| |P| |1
0|.|.|.|.|2
2|P|P| | |3
1|B|.| | |P4
      0 1 2 3 4

```

Indique a próxima jogada (Horizontal,Vertical): **_**

Esgotando-se a possibilidade de colocação de barcos na primeira coluna do tabuleiro, as casas sobran- tes foram marcadas como água. O jogo deve prosseguir desta forma até à colocação de todos os barcos nos respectivos portos. A qualquer momento, o jogador pode desistir do jogo, para tal deve indicar a instrução **CANCEL**. Pode igualmente, salvar o estado atual do jogo para retornar mais tarde, indicando a instrução **SAVE**. Em qualquer um destes casos, voltará ao menu principal da aplicação.

Menus do jogo

A aplicação deverá apresentar o seguinte menu:

```
Olá <Nome do Jogador>

1 - Iniciar Jogo
2 - Carregar Jogo
3 - Pontuações Pessoais
4 - Pontuações Gerais
5 - Criação de Níveis
0 - Sair
```

Figura 4 - Interface do menu principal de jogo

A opção Carregar Jogo permite voltar ao jogo anteriormente guardado. A opção Pontuações Pessoais deverá, relativamente ao jogador, apresentar a sua pontuação acumulada, bem como uma lista com as melhores pontuações obtidas, para cada nível. Já as Pontuações Gerais, deverá apresentar, de forma ordenada, a lista de pontuação acumulada por jogador.

```
Pontuações de <nome do jogador>

Pontuação Geral: <pontos>

<nível> - <pontos>
<nível> - <pontos>
<nível> - <pontos>
<nível> - <pontos>
<nível> - <pontos>

Prima qualquer tecla para voltar ao menu principal
```

Figura 5 - Interface da listagem de pontuações pessoais

```
Pontuações Gerais

1º - <nome do jogador> - <pontos>
2º - <nome do jogador> - <pontos>
3º - <nome do jogador> - <pontos>
4º - <nome do jogador> - <pontos>
5º - <nome do jogador> - <pontos>
6º - <nome do jogador> - <pontos>
7º - <nome do jogador> - <pontos>

Prima qualquer tecla para voltar ao menu principal
```

Figura 6 - Interface da listagem de pontuações gerais

Ao iniciar novo jogo deverão aparecer as seguintes opções:

```
1 - Iniciar jogo aleatório
2 - Iniciar jogo - Fácil
3 - Iniciar jogo - Médio
4 - Iniciar jogo - Difícil
0 - Voltar
```

Figura 7 - Interface do submenu de início de jogo

Se o jogador optar por iniciar um jogo aleatório, será sorteado um qualquer nível de todos os existentes. Ao seleccionar um determinado grau de dificuldade, deverá seleccionar, caso exista, um qualquer nível que corresponda a esse grau de dificuldade.

3. Considerações

3.1. Fases de desenvolvimento e entrega

O projeto está dividido em três fases, com a cotação distribuída da seguinte forma:

- 1ª Fase: análise e design, implementação das classes de domínio (30% da avaliação final);
- 2ª Fase: lógica do problema, implementação em ambiente de consola (40% da avaliação final);
- 3ª Fase: implementação em ambiente gráfico (30% da avaliação final).

É possível proceder a melhorias contínuas do projeto submetido, em determinada fase, na entrega seguinte. No entanto, todas as melhorias deverão ser identificadas no ficheiro “README”, que acompanha o projeto.

Conforme referido na ficha da disciplina, poderá alternativamente entregar o projeto numa só data, na época de recurso (sem possibilidade de incorporar componente de avaliação contínua), contemplando as três fases do projeto.

3.2. Implementação e codificação

O programa deve ser desenvolvido utilizando a linguagem Java, colocando em prática os conceitos fundamentais do paradigma de Programação Orientada por Objetos.

Em relação às regras de codificação, siga as convenções adotadas normalmente para a linguagem Java:

- A notação *camelCase* para o nome das variáveis locais e identificadores de atributos e métodos;
- A notação *PascalCase* para os nomes das classes e interfaces;
- Utilização de maiúsculas para os nomes das constantes e dos valores enumerados;
- Não utilize o símbolo ‘_’ nos identificadores (exceto nas constantes), nem abreviaturas.

É necessário que o projeto cumpra o que é pedido no seu enunciado, sendo deixado ao critério do programador qualquer pormenor de implementação que não seja referido, o qual deverá ser devidamente documentado.

3.3. Constituição de grupos

Cada projeto deverá ser elaborado em grupos de dois alunos, podendo excecionalmente ser elaborado de forma individual, em situações devidamente justificadas e aprovadas. Não serão permitidos, em nenhum caso, grupos com mais do que dois alunos.

Os grupos dos alunos já se encontram determinados através da metodologia de *pair programming* que está a ser utilizada nas aulas laboratoriais. Caso existam alunos que não tenham grupo formado, deverão contactar o respetivo docente de laboratório para regularizar a situação.

4. Entrega do projeto

O projeto será entregue em três fases distintas de implementação:

- Uma primeira fase (**até às 23 horas do dia 8 de maio de 2021**) com a implementação das classes de domínio da aplicação (descrita no presente documento);
- A segunda fase (**até às 23 horas do dia 29 de maio de 2021**) com a implementação em ambiente de consola (descrita no presente documento).
- A terceira fase (**até às 23 horas do dia 25 de junho de 2021**) com a implementação em ambiente gráfico (descrita em documento entregue futuramente).

O projeto deverá ser entregue até à data-limite especificada por **via exclusivamente eletrónica utilizando a plataforma GitHub**. Todos os ficheiros que compõem o projeto deverão estar submetidos no respetivo repositório, até à data estipulada. Em caso de dificuldades no acesso à plataforma GitHub, o envio dos ficheiros poderá ser feito por correio eletrónico para o respetivo docente de laboratório, dentro do prazo acima indicado.

Não serão aceites quaisquer projetos entregues fora do prazo!

Todos os materiais do projeto devem ser devidamente identificados com nome, número e endereço de correio eletrónico dos alunos.

Os materiais do projeto deverão incluir:

- Um **Manual Técnico** onde conste uma breve descrição do programa, incluindo a explicação das classes/interfaces implementadas, principais atributos e métodos e suas relações. Na primeira fase de implementação, deve ser realizado uma análise detalhada do desenho da aplicação, onde se pretende a inclusão de:
 - Análise de substantivos/verbos;
 - Criação de respetivas cartas CRC;
 - Diagrama de classes.
- A **documentação do programa em JavaDoc** (não converta o documento gerado automaticamente em HTML para DOC!).
- O **código fonte** do programa na forma de projeto em *NetBeans* ou *IntelliJ IDEA*, com um *main* de testes a funcionar (que instancie todos os objetos necessários à simulação) e com todas as funcionalidades implementadas.

5. Regras e Critérios de Avaliação do Projeto

5.1. Regras de Avaliação

A avaliação do projeto está sujeita às seguintes regras:

- **Não serão aceites quaisquer projetos entregues fora do prazo!**
- A classificação do programa terá em conta a qualidade da programação (fatores de qualidade do software), a estrutura do código criado segundo os princípios da Programação Orientada por Objetos, tendo em conta conceitos como a coesão de classes e métodos, o grau de acoplamento entre classes e o desenho de classes orientado pela responsabilidade, e a utilização/conhecimento da linguagem Java.
- Será valorizada a facilidade de utilização, a apresentação, a imaginação e a criatividade.
- O projeto terá uma componente de avaliação oral obrigatória com classificação individual dos elementos do grupo.
- Os alunos que não comparecerem à discussão serão classificados com zero na fase respetiva. Nesta discussão será apurada a capacidade do aluno de produzir o código apresentado. Nos casos em que essa capacidade não for demonstrada, a nota atribuída será zero.
- A avaliação oral é realizada pelo respetivo professor de laboratório e irá ser feita uma marcação prévia para cada grupo de trabalho.
- Todos os projetos serão submetidos a um sistema automático de deteção de cópias. Os projetos que forem identificados como possíveis cópias, e verificando-se serem realmente cópias, serão ambos anulados, quer o projeto original quer o plágio.
- As avaliações da primeira fase do projeto serão realizadas na segunda semana de maio de 2021.

5.2. Critérios de Avaliação

5.2.1. 1ª Fase

A primeira fase do projeto terá foco nas temáticas de: definição de classes, desenho de aplicações, herança de classes, polimorfismo e classes abstratas. Esta fase será avaliada segundo os seguintes critérios:

Funcionalidades	40%
------------------------	------------

Implementação	30%
Estrutura de classes	15%
Conhecimento e boa utilização da linguagem	5%
Bom estilo (nomes, comentários, indentação)	5%
Definição de testes unitários	5%

Documentação	20%
JavaDOC	5%
Manual técnico	15%

Avaliação qualitativa	10%
------------------------------	------------

5.2.2. 2ª Fase

A segunda fase do projeto terá foco na implementação da mecânica de jogo em ambiente de consola, na qual será arranjada as seguintes temáticas: coleções e genéricos, iteradores e processamento funcional, tratamento de erros e exceções, entrada e saída de dados (I/O) e serialização. Esta fase será avaliada segundo os seguintes critérios:

Funcionalidades	50%
------------------------	------------

Implementação	25%
Estrutura de classes	10%
Conhecimento e boa utilização da linguagem	5%
Bom estilo (nomes, comentários, indentação)	5%
Definição de testes unitários	5%

Documentação	10%
JavaDOC	5%
Manual técnico	5%

Avaliação qualitativa	15%
------------------------------	------------

Qualidade geral	10%
Componente criativa	5%

6. Resumo das Datas Importantes

6.1. Entrega da 1ª fase

A entrega da 1ª fase do projeto será até às 23 horas de sábado, dia 8 de maio de 2021.

6.2. Entrega da 2ª fase

A entrega da 2ª fase do projeto será até às 23 horas de sábado, dia 29 de maio de 2021.